Claim Amendments

Please amend the claims to be as follows.

1. (currently amended) A method of executing program code on a target microprocessor with multiple CPU cores thereon, the method comprising:
   selecting one of the CPU cores for testing;
   performing inter-core context switching;
   executing in parallel diagnostic code on the selected CPU core and the program code on remaining CPU cores,
   wherein the inter-core context switching includes swapping virtual CPU numbers between the CPU core selected for diagnostics and a recently-tested CPU core being put back to use.

2. (original) The method of claim 1, wherein the selection of the CPU core for testing utilizes an algorithm that assures testing of each of the multiple CPU cores.

3. (original) The method of claim 2, wherein the algorithm comprises a round-robin type algorithm.

4. (original) The method of claim 1, further comprising:
   setting a level of aggressiveness for scheduling the testing of the execution units.

5. (original) The method of claim 4, further comprising:
   applying an aggressiveness-dependent algorithm to determine when to schedule all available cores for execution of the program code and when to schedule parallel execution of the program code and the diagnostic code.

6. (original) The method of claim 1, wherein the multiple CPU cores comprise at least four CPU cores integrated onto the microprocessor integrated circuit.

7. (original) The method of claim 1, wherein the multiple CPU cores comprise at least eight CPU cores integrated onto the microprocessor integrated circuit.

8. (original) The method of claim 1, wherein the diagnostic code performs diagnostic operations from a test pattern comprising operations with known expected results.

9. (original) The method of claim 8, wherein the diagnostic code compares an actual result with a known expected result.

10. (original) The method of claim 9, wherein the diagnostic code jumps to a fault handler if the compared results are different.

11. (original) The method of claim 10, wherein the fault handler includes code to remove a faulty CPU core from use in executing the program code.

12. (original) The method of claim 10, wherein the fault handler includes code to perform a system halt to prevent data corruption.

Claims 13-14. (canceled)

15. (currently amended) A microprocessor comprising:

a plurality of CPU cores integrated on the microprocessor chip; and

inter-core communications circuitry coupled to each of the CPU cores and configured to perform context switching between the CPU cores,

control circuitry coupled to the inter-core communications circuitry and configured select a first CPU core currently in use for diagnostic testing,

wherein the inter-core communications circuitry is utilized to perform context switching between the first CPU core and a second CPU core which is not currently in use,

wherein the microprocessor is configured to swap external CPU numbers between the first and second CPU cores.

16. (original) The microprocessor of claim 15, wherein each CPU core comprises a processor core and an associated local cache memory.

Claims 17-20. (canceled)